1. (10 points) \Remove Element". Provide the code for the following question.

Given an integer array nums and an integer val, remove all occurrences of val in nums in-place. The order of the elements may be changed. Then return the number of elements in nums which are not equal to val.

Consider the number of elements in nums which are not equal to val be k, to get accepted, you need to do the following things:

Change the array nums such that the first k elements of nums contain the elements which are not equal to val. The remaining elements of nums are not important as well as the size of nums.

**Example 1:**

Input: nums = [3,2,2,3], val = 3

Output: 2, nums = [2,2,\_,\_]

Explanation: Your function should return k = 2, with the first two elements of $nums$ being $2$. It does not matter what you leave beyond the returned k (hence they are underscores).

**Example 2:**

Input: nums = [0,1,2,2,3,0,4,2], val = 2

Output: 5, nums = [0,1,4,0,3,\_,\_,\_]

Explanation: Your function should return k = 5, with the first five elements of nums containing 0, 0, 1, 3, and 4.

Note that the five elements can be returned in any order. It does not matter what you leave beyond the returned k (hence they are underscores).

**Code:**

def removeElement(nums, val):  
 write\_index = 0

for read\_index in range(len(nums)):

if nums[read\_index] != val:

nums[write\_index] = nums[read\_index]

write\_index += 1

return write\_index

1. Non-decreasing array". Provide your explanation for a solution to the following problem. You do not need to give me the code.

Given an array nums with n integers, your task is to check if it could become non-decreasing by modifying at most one element.

We define an array is non-decreasing if nums[i] <= nums[i + 1] holds for every i (0-based) such that (0 <= i <= n - 2).

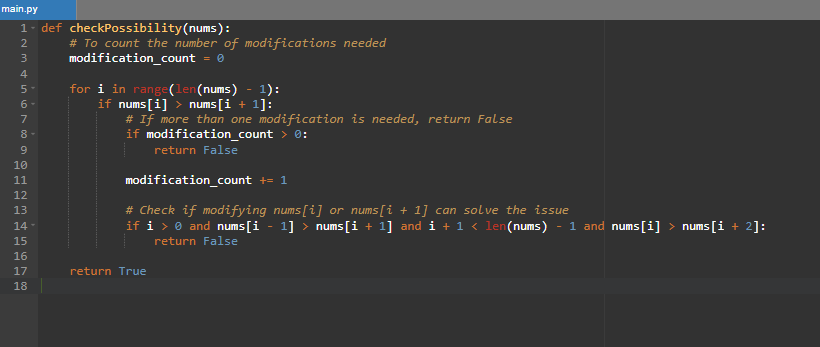
**Example 1:**

Input: nums = [4,2,3]

Output: true

Explanation: You could modify the first 4 to 1 to get a non-decreasing array.

**Code:**

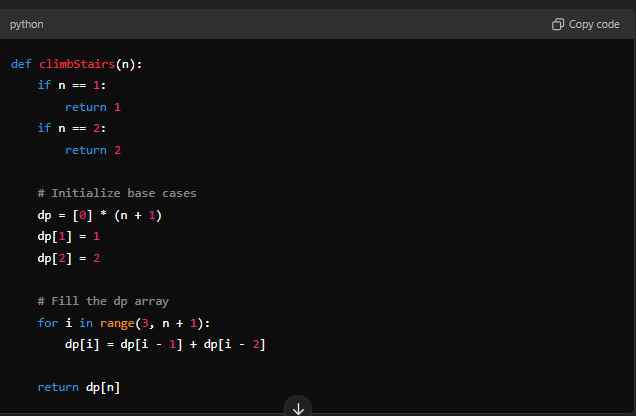
****

1. (10 points) \Climbing Stairs". Provide the code for the following question.

You are climbing a staircase. It takes n steps to reach the top.  
Each time you can either climb 1 or 2 steps.  
In how many distinct ways can you climb to the top?

Example 1:

Input: n = 2  
Output: 2  
Explanation: There are two ways to climb to the top.  
1. 1 step + 1 step  
2. 2 steps



1. (10 points) \-Merge k- Sorted Lists". Provide your explanation for a solution to the following problem. You do not need to give me the code.

You are given an array of k linked-lists lists, each linked-list is sorted in ascending order.

Merge all the linked-lists into one sorted linked-list and return it.

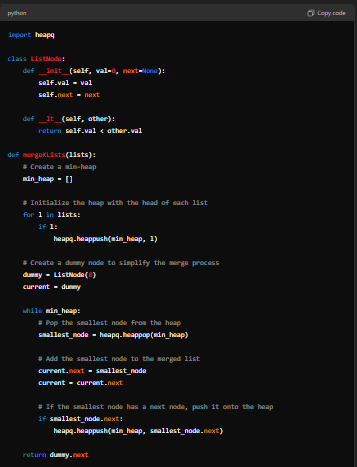
Example 1:

Input: lists = [[1,4,5],[1,3,4],[2,6]]

Output: [1,1,2,3,4,4,5,6]

Explanation: The linked-lists are:

[  
1->4->5,  
1->3->4,  
2->6  
]  
merging them into one sorted



1. (10 points) \Delete Node in a BST". Provide your explanation for a solution to the following problem. You do not need to give me the code.

Given a root node reference of a BST and a key, delete the node with the given key in the BST. Return the root node reference (possibly updated) of the BST.  
Basically, the deletion can be divided into two stages:

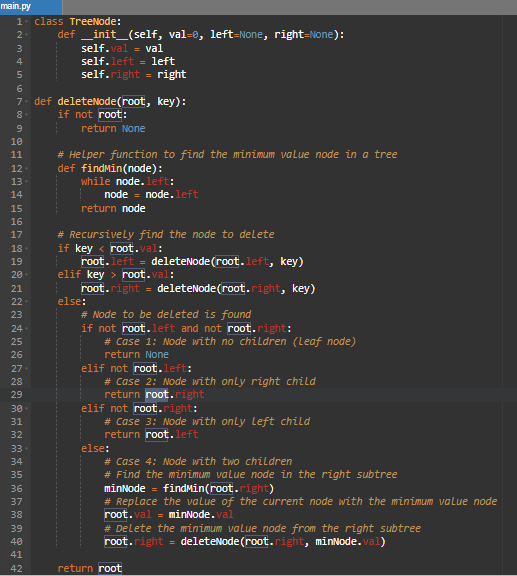
- Search for a node to remove.  
- If the node is found, delete the node.

Example 1:  
- Input: root = [5,3,6,2,4,null,7], key = 3  
- Output: [5,4,6,2,null,null,7]

Explanation: Given key to delete is 3. So we find the node with value 3 and delete it.

One valid answer is [5,4,6,2,null,null,7], shown in the above BST.

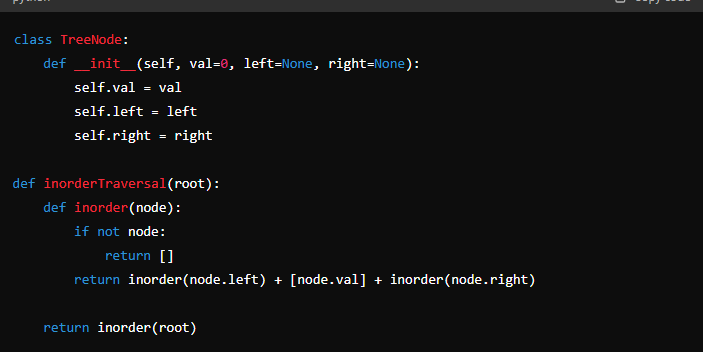
Please notice that another valid answer is [5,2,6,null,4,null,7] and it's also accepted.

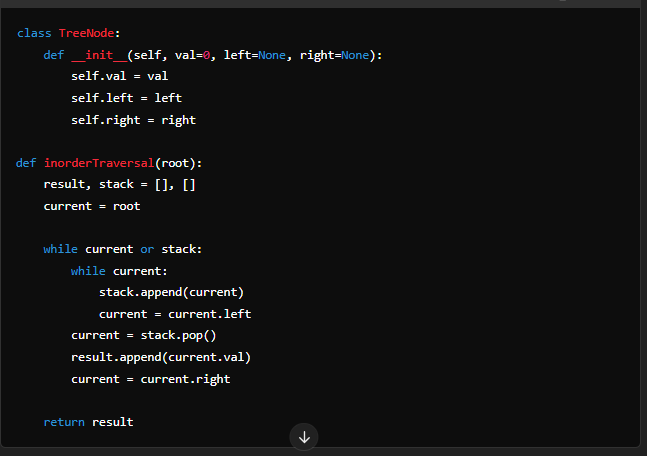


1. (10 points) \Binary Tree Inorder Traversal". Provide the code for the following question.

Given the root of a binary tree, return the inorder traversal of its nodes' values.

\*\*Example 1:\*\*  
- Input: root = [1,null,2,3]  
- Output: [1,3,2]



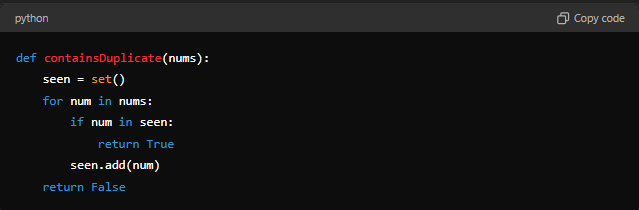


1. (10 points) \Contains Duplicate". Given an integer array nums, return true if any value appears at least twice in the array, and return false if every element is distinct.

- Example 1:

- Input: nums = [1,2,3,1]

- Output: true



1. (10 points) \LRU Cache-Medium". Design a data structure that follows the constraints of a Least Recently Used (LRU) cache.

Implement the LRUCache class:  
LRUCache(int capacity) Initialize the LRU cache with positive size capacity.  
int get(int key) Return the value of the key if the key exists, otherwise return -1.  
void put(int key, int value) Update the value of the key if the key exists.  
Otherwise,add the key-value pair to the cache. If the number of keys exceeds  
the capacity from this operation, evict the least recently used key.

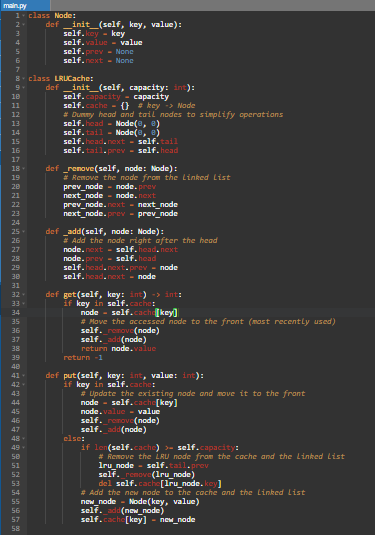
The functions get and put must each run in O(1) average time complexity.

- Example 1:

- Input ["LRUCache", "put", "put", "get", "put", "get", "put", "get", "get", "get"] [[2], [1, 1], [2, 2], [1], [3, 3], [2], [4, 4], [1], [3], [4]]

- Output: [null, null, null, 1, null, -1, null, -1, 3, 4]

- Explanation  
- LRUCache lRUCache = new LRUCache(2);  
- lRUCache.put(1, 1); // cache is {1=1}  
- lRUCache.put(2, 2); // cache is {1=1, 2=2}  
- lRUCache.get(1); // return 1  
- lRUCache.put(3, 3); // LRU key was 2, evicts key 2, cache is {1=1, 3=3}  
- lRUCache.get(2); // returns -1 (not found)  
- lRUCache.put(4, 4); // LRU key was 1, evicts key 1, cache is {4=4, 3=3}  
- lRUCache.get(1); // return -1 (not found)  
- lRUCache.get(3); // return 3  
- lRUCache.get(4); // return 4



1. (10 points) \Course Schedule". There are a total of numCourses courses you have to take, labeled from 0 to numCourses - 1.   
   You are given an array prerequisites where prerequisites[i] = [ai, bi] indicates that you must take course bi first if you want to take course ai.

For example, the pair [0, 1], indicates that to take course 0 you have to first take course 1.   
Return true if you can finish all courses. Otherwise, return false.

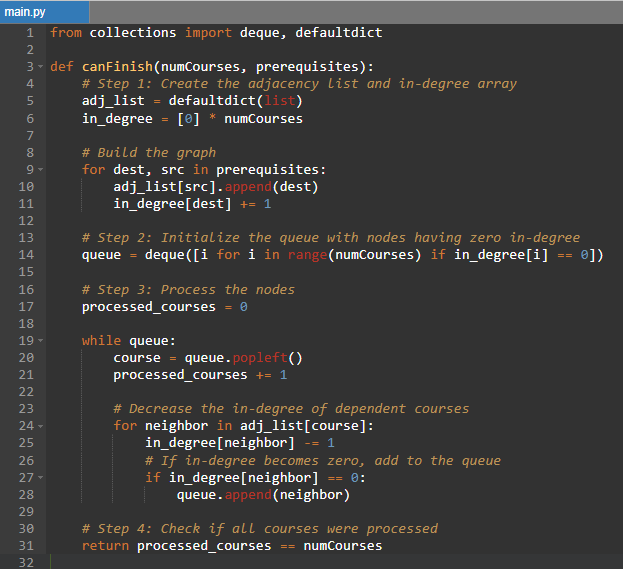
- Example 1:

- Input: numCourses = 2, prerequisites = [[1,0]]

- Output: true

- Explanation: There are a total of 2 courses to take. \

To take course 1 you should have finished course 0. So it is possible.



1. (10 points) \Course Schedule II". There are a total of numCourses courses you have to take, labeled from 0 to numCourses - 1. You are given an array prerequisites where prerequisites[i] = [ai, bi] indicates that you must take course bi first if you want to take course ai.

For example, the pair [0, 1], indicates that to take course 0 you have to first take course 1. Return the ordering of courses you should take to finish all courses. If there are many valid answers, return any of them. If it is impossible to finish all courses, return an empty array.

- Example 1:

- Input: numCourses = 2, prerequisites = [[1,0]]  
- Output: [0,1]  
- Explanation: There are a total of 2 courses to take.\  
To take course 1 you should have finished course 0.  
So the correct course order is [0,1].

